第九节 服务调用

基本概念

消息（Message）

主要部分：消息头（Headers）和消息体（Body）

Headers = Metadata（元数据）

Body = Data（主要数据）

存根（Stub）

大致上等于 Proxy、Handler，通常是基于 Java 接口编程，JDK 动态代理

Servlet 请求与线程映射关系

ServletRequest 和 Thread 一对一的映射关系

ServletRequestListener 可以在 requestInitialized 方法中将对象保存在 ThreadLocal 中，

requestDestroyed 方法中移除 ThreadLocal 引用

Servlet 容器（引擎）线程实现基于线程池，当请求消亡后，但是线程还在复用

REST 规范

URI 范例

```
http://www.baidu.com/a/b/c?q=mercyblitz#p
```

scheme = http

host = www.baidu.com

port = 80

path = /a/b/c

query = "q=mercyblitz"

fragment(锚点) = p

schemeSpecificPart

REST 请求基本步骤

1. 构建 URI - 请求资源
2. 确定请求方法 - GET、POST
3. 设置请求头和参数 - Headers、Parameters
4. 设置请求主题（可选） - Body
5. URI -> 设置到请求
6. 执行请求（发送到 Server 服务器）
7. 处理响应
    1. 正确响应（200，2XX）
    2. 异常响应
        1. 请求有误（4XX）
        2. 服务器问题（5XX）
        3. 请求转移（3XX）

JAX-RS 规范

应用

- 基于 Servlet Container
- 基于 Standalone（独立）

处理步骤

1. 构建 URI - 请求资源
    - UriBuilder
2. 确定请求方法 - GET、POST
    - @HttpMethod
3. 设置请求头和参数 - Headers、Parameters
    - Header Name 和 Header Value（多值）
    - Parameter Name 和 Parameter Value（多值）
    - 数据结构：`Map<String,List<String>>` - MultivaluedMap
4. 设置请求主体（可选） - Body
    - 二进制流
        - 可以转换为 Reader
5. URI -> 设置到请求
    - HTTP 客户端发送请求
        - JDK HttpURLConnection
        - Apache HttpClient 3.x
        - Apache HttpComponents
        - OkHttp
6. 执行请求（发送到 Server 服务器）

    - Servlet Stack（Tomcat、Jetty、Undertown）
    - Spring WebFlux（Netty Web Server）
    - Vert.x
7. 处理响应
    1. 正确响应（200，2XX）
        - 状态码
        - 响应头
            - 数据结构： `Map<String,List<String>>`
        - 响应主体
            - 二进制
    2. 异常响应 - ExceptionMapper
        1. 请求有误（4XX）
        2. 服务器问题（5XX）
        3. 请求转移（3XX）

API

Application

RuntimeDelegate

UriBuilder

构建 Java 标准的 java.net.URI

想要构建 URI -> UriBuilder#build() -> 前提获取 UriBuilder 实例

UriBuilder的静态方法获取 UriBuilder 实例

UriBuilder#newInstance() -> RuntimeDelegate.getInstance().createUriBuilder()

服务端

客户端

实现步骤

1. 创建客户端 - Client
    - 将 javax.ws.rs.client.ClientBuilder 类按照 Java SPI 实现
    - 获取 ClientBuilder 实例 - javax.ws.rs.client.ClientBuilder#newBuilder
    - ClientBuilder#newClient()
    - 实现 javax.ws.rs.client.Client
        - 实现 target 方法
            - 实现 RuntimeDelegate#createUriBuilder()
            - 实现 WebTarget
2. 获取 Web 资源目标 - WebTarget
3. 执行请求 - Invocation.Builder extends SyncInvoker
    - 依赖资源
        - URI - UriBuilder
        - 请求方法 - GET、POST 等
        - 请求参数（Query String）、请求头（Headers）、请求主体（Body OutputStream）
4. 获取响应对象 - Response
    - Invocation -> SyncInvoker.invoke()

请求头

```
Accept:
text/html,application/xhtml+xml,application/xml
;q=0.9,image/avif,image/webp,image/apng,*/*;q=0
.8,application/signed-exchange;v=b3;q=0.9
```

```
Cookie:
JSESSIONID=FA0A49FD17A95D9E76F37492A87B2FAD;
JSESSIONID.41c5af47=node01ahqm3hwiug512o8bzp8v0
h8i0.node0;
JSESSIONID.74dead16=node01vhwyfbvrryrzr3vgqxofc
hck0.node0;
JSESSIONID.8c51d745=node01a840hp50718g198dg1dxn
6sw60.node0;
JSESSIONID.ee530a81=node0ys2dax1lx9b0wpamj2aml9
dp2.node0; screenResolution=1920x1080; jenkins-
timestamper-offset=-28800000;
JSESSIONID.36710276=node05hhgtk7zjj0z1dacf9jpn8
oa0.node0
```

基于 JDK HttpURLConnection

Jackson

ObjectMapper -> InputStream -> POJO